Spam filter

Algoritmusok tervezése és elemzése

A black background with blue text

Description automatically generated

Készítette: Stumpf Máté Benedek

Konzulens: Dr. Túri József PhD

Tartalomjegyzék:

[***Pszeudokód:*** 2](#_Toc185540297)

[***A program forráskódja:*** 4](#_Toc185540298)

[***Itt található pár példa a kimenetre:*** 6](#_Toc185540299)

[***Végszó:*** 7](#_Toc185540300)

A problémám a manapság egyre gyakoribb spam emailek és online csalások megelőzése. E célból készítettem egy spam szűrő programot, amely bemenetként fogad egy akár több soros email szöveget. Ez a szöveg bemásolható a terminálba (***másolás után nyomjuk meg a Ctrl + Z, vagy a Ctrl + D billentyűkombinációt, futtatási környezettől függően***), ahol a program elemzi, hogy tartalmaz-e előre meghatározott spam-re utaló kulcsszavakat vagy URL-eket.

A program feldolgozza az email tartalmát, a sorvégeket szóközökre cseréli, majd kereső algoritmusokkal azonosítja a kulcsszavakat és az URL-eket. Az eredményeket megjeleníti: felsorolja a kulcsszavakat azok előfordulási számával együtt, valamint a megtalált URL-eket, és ellenőrzi, hogy azok tanúsítványa biztonságos-e (a „http://” használata alapján).

A program végül eldönti, hogy az email spam-e, és erről tájékoztatja a felhasználót. A kulcsszókereséshez, az URL-ek felismeréséhez és az eredmények kiértékeléséhez kereső, eldöntő és megszámláló algoritmusokat alkalmaztam.

Itt található a pszeudokód, és azt követően majd a program forráskódja, Python nyelven. (A tartalomjegyzék hyperlinkeket tartalmaz, hogy megkönnyítse a fejezetek közti keresést.)

***Pszeudokód:***

**Eljárás Bemenet\_Olvasása()**

**Írd ki: "Kérlek, másold be a szöveget (több soros is lehet). Ha végeztél, nyomd meg a Ctrl+D-t vagy a Ctrl+Z-t:"**

**Olvasd be az összes sort a standard bemenetről**

**Távolítsd el az új sorokat, helyettesítsd őket szóközökkel**

**Adj vissza a megtisztított szöveget**

**Eljárás vége**

Ez az eljárás a felhasználótól kéri a bemeneti adatot, amely lehet több soros szöveg. Eltávolítja a sorvégeket, és a sorokat egyetlen folyamatos szöveggé alakítja, amelyet visszaad a további feldolgozáshoz.

**Eljárás Spam\_Szűrő(szöveg)**

**Alakítsd át a szöveget kisbetűssé**

**Inicializáld TALÁLT\_KULCSSZAVAK-at, mint üres szótár**

**Inicializáld TALÁLT\_URL-eket, mint üres lista**

**Minden egyes kulcsszóra a SPAM\_KULCSSZAVAK-ban**

**Számold meg a kulcsszó előfordulásait a szövegben**

**Ha az előfordulások száma > 0 akkor**

**Add hozzá a kulcsszót és az előfordulások számát a TALÁLT\_KULCSSZAVAK-hoz**

**Elágazás vége**

**Ciklus vége**

**Keress meg minden egyezést az URL\_MINTA alapján a szövegben**

**Add hozzá az egyezéseket a TALÁLT\_URL-ekhez**

**Inicializáld NEM\_BIZTONSÁGOS\_URL-t, mint üres lista**

**Minden URL-re a TALÁLT\_URL-ekben**

**Ha az URL "http://"-el kezdődik akkor**

**Add hozzá az URL-t a NEM\_BIZTONSÁGOS\_URL-hez**

**Elágazás vége**

**Ciklus vége**

**Ellenőrizd, hogy a TALÁLT\_KULCSSZAVAK vagy a TALÁLT\_URL nem üresek-e**

**Adj vissza (SPAM\_E, TALÁLT\_KULCSSZAVAK, TALÁLT\_URL, NEM\_BIZTONSÁGOS\_URL)**

**Eljárás vége**

Ez az eljárás a bemeneti szöveget elemzi:

1. Átalakítja kisbetűssé az egységesítés érdekében.
2. Ellenőrzi, hogy tartalmaz-e előre meghatározott spam kulcsszavakat, és megszámolja az előfordulásokat.
3. Keres gyanús URL-eket a szövegben regex segítségével, és listázza őket.
4. Azonosítja a nem biztonságos URL-eket (http:// protokollal kezdődőket).
5. Visszaadja az eredményeket, beleértve azt is, hogy SPAM-nek minősíthető-e a szöveg.

**Eljárás Fő\_Menü()**

**Hívd meg a Bemenet\_Olvasása() eljárást, és tárold az eredményt a SZÖVEG változóban**

**Hívd meg a Spam\_Szűrő(SZÖVEG) eljárást, és tárold az eredményeket (SPAM\_E, TALÁLT\_KULCSSZAVAK, TALÁLT\_URL, NEM\_BIZTONSÁGOS\_URL)-ként**

**Ha SPAM\_E akkor**

**Írd ki: "Az email valószínűleg SPAM!"**

**Különben**

**Írd ki: "Az email valószínűleg NEM SPAM."**

**Elágazás vége**

**Ha TALÁLT\_KULCSSZAVAK nem üresek akkor**

**Írd ki: "Talált spam-re utaló kulcsszavak és előfordulásaik:"**

**Minden kulcsszó és szám párra a TALÁLT\_KULCSSZAVAK-ban**

**Írd ki a kulcsszót és az előfordulások számát**

**Ciklus vége**

**Elágazás vége**

**Ha TALÁLT\_URL nem üresek akkor**

**Írd ki: "Talált gyanús URL-ek:"**

**Minden URL-re a TALÁLT\_URL-ekben**

**Írd ki az URL-t**

**Ciklus vége**

**Elágazás vége**

**Ha NEM\_BIZTONSÁGOS\_URL nem üresek akkor**

**Írd ki: "Figyelmeztetés: A következő URL-ek tanúsítványa nem biztonságos:"**

**Minden URL-re a NEM\_BIZTONSÁGOS\_URL-ben**

**Írd ki az URL-t**

**Ciklus vége**

**Elágazás vége**

**Eljárás vége**

Ez az eljárás összehozza az összes részegységet:

1. Meghívja a bemenet olvasására szolgáló eljárást és eltárolja a szöveget.
2. Meghívja a spam szűrőt az elemzéshez.
3. Kiértékeli az eredményeket:
   * Jelzi, hogy a szöveg SPAM-e.
   * Külön listázza a talált kulcsszavakat és azok előfordulását.
   * Felsorolja a talált URL-eket.
   * Figyelmeztet a nem biztonságos URL-ekre.

**Hívd meg a Fő\_Menü() eljárást**

**Eljárás vége**

A program ezzel az utasítással indul el. Meghívja a Fő\_Menü eljárást, amely elvégzi a teljes folyamatot: beolvasás, elemzés és eredmények kiírása.

***A program forráskódja:***

import re

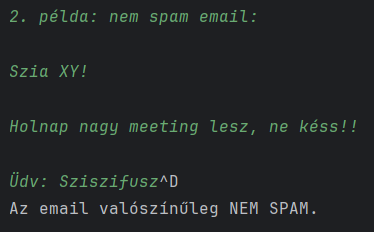
import sys  
  
# Gyakori spam kulcsszavak és kifejezések  
spam\_kulcsszavak = ["nyeremény","nyereményjáték", "ingyen", "kattints ide", "ajánlat", "gyors pénz", "növeld a bevételed", "pénzkeresési lehetőség"]  
  
# URL-ek keresésére használt minta  
url\_minta = r"http[s]?://(?:[a-zA-Z]|[0-9]|[$-\_@.&+]|[!\*\\(\\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+|(?:[a-zA-Z0-9-]+\.)+[a-zA-Z]{2,}"  
  
def bemenet\_olvasasa():  
 #Beolvassa a teljes bemenetet és megtisztítja.  
 print("Kérlek, másold be a szöveget (több soros is lehet). Ha végeztél, nyomd meg a Ctrl+D-t (pycharm IDE) vagy a Ctrl+Z-t:")  
 bemenet = sys.stdin.read() # Beolvassa az összes sort nyers bemenettel, így tudunk több sort egyszerre bevinni  
 return bemenet.strip().replace("\n", " ") # A sortöréseket szóközzé tesszük  
  
def spam\_filter(szoveg):  
  
 #Ellenőrzi, hogy a szöveg spam-e, és visszaadja a talált gyanús elemeket (kulcsszavak és URL-ek).  
  
 szoveg = szoveg.lower() # Kisbetűsre alakítjuk a szöveget  
 talalt\_kulcsszavak = {} #kulcsszavak tárolásához szükséges lista  
  
  
 # Kulcsszavak keresése és a megjelenésük megszámlálása  
 for kulcsszo in spam\_kulcsszavak:  
 elofordulas = szoveg.count(kulcsszo)  
 if elofordulas > 0:  
 talalt\_kulcsszavak[kulcsszo] = elofordulas  
  
 # URL-ek keresése  
 talalt\_url = re.findall(url\_minta, szoveg)  
  
 # Ellenőrizzük, hogy van-e 'http' URL és figyelmeztessünk a nem biztonságos tanúsítványra  
 nem\_biztonsagos\_url = [url for url in talalt\_url if url.startswith("http://")]  
  
 # A szöveg spam, ha találunk kulcsszavakat vagy URL-eket  
 is\_spam = bool(talalt\_kulcsszavak or nem\_biztonsagos\_url)  
  
 return is\_spam, talalt\_kulcsszavak, talalt\_url, nem\_biztonsagos\_url  
  
def main():  
 # Bemenet beolvasása és tisztítása  
 szoveg = bemenet\_olvasasa()  
  
 # Spam szűrés  
 is\_spam, talalt\_kulcsszavak, talalt\_url, nem\_biztonsagos\_url = spam\_filter(szoveg)  
  
 # Eredmény kiírása  
 if is\_spam:  
 print("Az email valószínűleg SPAM!")  
 else:  
 print("Az email valószínűleg NEM SPAM.")  
  
 # Talált kulcsszavak és URL-ek listázása  
 if talalt\_kulcsszavak:  
 print("Talált spam-re utaló kulcsszavak és előfordulásaik:")  
 for kulcsszo, db in talalt\_kulcsszavak.items():  
 print(f"- {kulcsszo}: {db} alkalommal")  
  
 #itt az URL-eket akkor listázza ki ha az email spamnek minősül  
 if talalt\_url and is\_spam:  
 print("Talált gyanús URL-ek:")  
 for url in talalt\_url:  
 print(f"- {url}")  
  
 # Nem biztonságos http URL-ek figyelmeztetése  
 if nem\_biztonsagos\_url:  
 print("Figyelmeztetés: A következő URL-ek tanúsítványa nem biztonságos:")  
 for url in nem\_biztonsagos\_url:  
 print(f"- {url}")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

# ***Itt található pár példa a kimenetre:***

A screen shot of a computer

Description automatically generated

***Nem spam email-re adott kimenet:***

******

# 

# ***Végszó:***

Azért ezt a problémát választottam, mivel engem főképpen a kiberbiztonság érdekel, és a későbbiekben ebben a szférában szeretnék elhelyezkedni.  
  
A programot a PyCharm nevű IDE-ben fejlesztettem.

***Github link a projekthez:***

<https://github.com/Y0r0zuya/Spam_Filter>

***Készítette: Stumpf Máté Benedek – HWS19X  
Tantárgy: Algoritmusok tervezése és elemzése  
Kurzusfelelős: Dr. Túri József***